REMARKS

Claims 1-7, 9-21 and 23-28 are pending in the application.

Claims 1-7, 9-21 and 23-28 have been rejected.

Claim 1 has been amended as set forth herein, solely to correct a typographical error.

Claims 1-7, 9-21 and 23-28 remain pending in this application.

Reconsideration of the claims is respectfully requested.

I. CLAIM REJECTIONS -- 35 U.S.C. § 103

Claims 1-3, 6-7, 9-10, 13-17, 20-21, 23-24, 27 and 28 were rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,094,715 to *Wilkinson, et al.* (hereinafter "Wilkinson") in view of U.S. Patent No. 4,435,758 to *Lorie, et al.* (hereinafter "Lorie") and further in view of U.S. Patent No. 6,823,517 to *Kalman* (hereinafter "Kalman"). Claims 4, 5, 18 and 19 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Wilkinson in view of Lorie, and further in view of Kalman, and further in view of U.S. Patent No. 6,470,441 to *Pechanek, et al.* (hereinafter "Pechanek"). Claims 11, 12, 25 and 26 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Wilkinson in view of Lorie, and further in view of Kalman, and further in view of Hansoo Kim "Multi-thread VLIW processor architecture for HDTV decoding" (hereinafter "Kim"). The Applicant respectfully traverses the rejections.

In *ex parte* examination of patent applications, the Patent Office bears the burden of establishing a prima facie case of obviousness. MPEP § 2142, p. 2100-133 (8th ed. rev. 4, October 2005). Absent such a prima facie case, the applicant is under no obligation to produce evidence of

nonobviousness. *Id.* To establish a *prima facie* case of obviousness, three basic criteria must be met: *Id.* First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. *Id.* Second, there must be a reasonable expectation of success. *Id.* Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *Id.* The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *Id.*

First, the Office Action asserts that the broadcast and control interface (BCI) in Wilkinson's processor memory element (PME) describes a job buffer. The BCI buffers individual instructions and data items for broadcast to PMEs. *See col. 24, lines 9-16 and 35-55.* On page 27, lines 14-15, of the Specification, a job is defined as a combination of a program and an input data-set. Thus, the Applicant respectfully submits that a person of skill in the art would not understand Wilkinson's BCI buffer for individual instructions and data words as teaching a buffer for jobs, as taught in the present application.

Second, the Office Action asserts that Wilkinson teaches a job status comprising a program counter value and a loop-counter list, wherein a job is a combination of a program and an input data-set, citing the following passages:

> The BCI is the node's interface to the external array controller element and to an array director. The BCI provides common node functions such as timers and clocks. The BCI provides broadcast function masking for each nodal PME and provides the physical interface and buffering for the broadcast-bus-to-PME data transfers, and also

provides the nodal interface to system status and monitoring and debug elements. *Col. 24, lines 9-17.*

Each PME is capable of operating in SIMD or in MIMD mode in our preferred embodiment. In SIMD mode, each instruction is fed into the PME from the broadcast bus via the BCI. The BCI buffers each broadcast data word until all of its selected nodal PMEs have used it. This synchronization provides accommodation of the data timing dependencies associated with the execution of SIMD commands and allows asynchronous operations to be performed by a PME. In MIMD mode, each PME executes its own program from its own main store. The PMEs are initialized to the SIMD mode. For MIMD operations, the external controller normally broadcasts the program to each of the PMEs while they are in SIMD mode, and then commands the PMEs to switch to MIMD mode and begin executing. Masking/tagging the broadcast information allows different sets of PMEs to contain different MIMD programs, and/or selected sets of PMEs to operate in MIMD mode while other sets of PMEs execute in SIMD mode. In various software clusters or partitions these separate functions can operate independently of the actions in other clusters or partitions. *Col. 24, lines 34-47.*

The BCI broadcast and control interface provided on each chip provides a parallel input interface such that data or instructions can be sent to the node. Incoming data is tagged with a subset identifier and the BCI includes the functions required to assure that all PMEs within the node, operating within the subset, are provided the data or instructions. The parallel interface of the BCI serves both as a port to permit data to be broadcast to all PMEs and as the instruction interface during SIMD operations. Satisfying both requirements plus extending those requirements to supporting subset operations is completely unique to this design approach.

Our BCI parallel input interface permits data or instructions to be sent from a control element that is external to the node. The BCI contains "group assignment" registers (see the grouping concepts in our above application entitled GROUPING OF SIMD PICKETS) which are associated with each of the PMEs. Incoming data words are tagged with a group identifier and the BCI includes the functions required to assure that all PMEs within the node which are assigned to the dedicated group are provided the data or instructions. The parallel interface of the BCI serves as both a port to permit data to be broadcast to the PMEs during MIMD operations, and as the PME instruction/immediate operand interface during SIMD operations.
Col. 33, line 51, through col. 34, line 7.

The major parts of the internal data flow of the processing element are shown in FIG. 7. FIG. 7 illustrates the internal data flow of the processing element. This processing element has a full 16 bit internal data flow 425, 435, 445, 455, 465. The

important paths of the internal data flows use 12 nanosecond hard registers such as the OP register 450, M register 440, WR register 470, and the program counter PC register 430. These registers feed the fully distributed ALU 460 and I/O router registers and logic 405, 406, 407, 408 for all operations. With current VLSI technology, the processor can execute memory operations and instruction steps at 25 Mhz, and it can build the important elements, OP register 450, M register 440, WR register 470, and the PC register 430 with 12 nanosecond hard registers. Other required registers are mapped to memory locations.
Col. 26, lines 6-21.

The interrupt levels are assigned to the input ports, the BCI, and to error handling. There is a "normal" level, but there is no "privileged", nor "supervisor" level. A program interrupt causes a context switch in which the contents of the PC program counter, status/control register, and selected general registers are stored in specified main memory locations and new values for these registers are fetched from other specified main memory locations.
Col. 29, lines 44-52.

To enable the network to be efficient for all types of transfer requirements, we partition, between the H/W and S/W, the responsibility for data routing between PMEs. S/W does most of the task sequencing function. We added special features to the hardware (H/W) to do the inner loop transfers and minimize software (S/W) overhead on the outer loops.
Col. 39, lines 60-65.

The PME is initialized to SIMD mode with interrupts disabled. Commands are fed into the PME operation decode buffer from the BCI. Each time an instruction operation completes, the PME requests a new command from the BCI. In a similar manner, immediate data is requested from the BCI at the appropriate point in the instruction execution cycle. Most instructions of the ISA operate identically whether the PME is in SIMD mode or in MIMD mode, with the exception of that SIMD instructions and immediate data are taken from the BCI; in MIMD mode the PME maintains a program counter (PC) and uses that as the address within its own memory to fetch a 16 bit instruction. Instructions such as "Branch" which explicitly address the program counter have no meaning in SIMD mode and some of those code points are reinterpreted to perform special SIMD functions as comparing immediate data against an area of main store.
Col. 27, lines 6-21.

The Applicant notes that all references in the cited passages to a program counter are related to the PME operating in MIMD mode. In fact, when discussing the memory of the PME, Wilkinson makes clear that in SIMD mode all the PME memory is used for data. *See col. 29, lines 8-12.* As such, a person of skill in the art would not find in Wilkinson a teaching of a job status comprising a program counter value in the context of a job buffer for a SIMD unit.

Third, the Office Action acknowledges that Wilkinson does not specifically disclose a job status comprising a loop-counter list, but asserts that it would have been obvious to a person of skill in the art to have a loop-count list as status of a task. However, such an argument fails to consider the invention recited in Claim 1 as a whole: i.e., using a job status that includes a program counter value and a loop-counter list to determine an equivalence of jobs. Furthermore, it is unclear whether the Office Action is arguing that a loop-counter list is an inherent component of the status of a task or is taking Official Notice of such a purported fact.

If the Office Action is arguing inherency, then the Office Action has failed to establish that a list of loop-counters is necessarily present in the SIMD processors of Wilkinson. As discussed above, Wilkinson's SIMD processors do not teach program counters. Therefore, the Applicant respectfully submits that it is improper to suggest that loop-counter lists are necessarily present in Wilkinson.

The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. *In re Rijckaert*, 9 F.3d 1531, 1534, 28 USPQ2d 1955, 1957 (Fed. Cir. 1993). "To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference,

and that it would be so recognized by persons of ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.' " *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950-51 (Fed. Cir. 1999) (citations omitted, emphasis added).

If the Office Action is taking Official Notice that it is common knowledge that a job status includes a list of loop-counters, the Applicant respectfully submits that such a assertion is not capable of instant and unquestionable demonstration, as required under M.P.E.P. § 2144.03(A):

> Official notice unsupported by documentary evidence should only be taken by the examiner where the facts asserted to be well-known, or to be common knowledge in the art are capable of instant and unquestionable demonstration as being well-known. As noted by the court in In re Ahlert, 424 F.2d 1088, 1091, 165 USPQ 418, 420 (CCPA 1970), the notice of facts beyond the record which may be taken by the examiner must be "capable of such instant and unquestionable demonstration as to defy dispute" (citing In re Knapp Monarch Co., 296 F.2d 230, 132 USPQ 6 (CCPA 1961)).
> * * *
> It would not be appropriate for the examiner to take official notice of facts without citing a prior art reference where the facts asserted to be well known are not capable of instant and unquestionable demonstration as being well-known.
> * * *
> It is never appropriate to rely solely on "common knowledge" in the art without evidentiary support in the record, as the principal evidence upon which a rejection was based. Zurko, 258 F.3d at 1385, 59 USPQ2d at 1697 ("[T]he Board cannot simply reach conclusions based on its own understanding or experience-or on its assessment of what would be basic knowledge or common sense. Rather, the Board must point to some concrete evidence in the record in support of these findings."). While the court explained that, "as an administrative tribunal the Board clearly has expertise in the subject matter over which it exercises jurisdiction," it made clear that such "expertise may provide sufficient support for conclusions [only] as to peripheral issues." Id. at 1385-86, 59 USPQ2d at 1697. As the court held in Zurko, an assessment of basic knowledge and common sense that is not based on any evidence in the record lacks substantial evidence support. Id. at 1385, 59 USPQ2d at 1697.

Because the presence of a loop-counter list in a job status is not a peripheral issue and because such a presence is not "capable of such instant and unquestionable demonstration as to defy dispute," it is improper for the Office Action to take official notice that it would have been obvious to a person of skill in the art to have a loop-count list as status of a task.

Fourth, the Office Action acknowledges that Wilkinson does not teach a job buffer dynamically bundling jobs into a task based on an equivalence of a job status of the jobs and allocating said task to a SIMD unit. However, the Office Action asserts that Lorie teaches such dynamic bundling in column 1, lines 32-45:

> However, if as in many data base applications, there is a high degree of contention for resources but the transactions are extremely homogeneous, then a network of synchronous processors working in an SIMD (single instruction multiple data) mode may be indicated. Groups of similar tasks may then be batched and run together through such a processor, synchronization minimizing the interprocessor communication is necessary in order to manage the resource contention. If the task consists of streams of straight line code (no branches), then all that is needed is a special purpose operating system for grouping, loading relevant data, starting and stopping.

That is, prior to execution, streams of straight line code may be recognized in a task and a special purpose operating system used to group, load relevant data, start and stop such streams of code.

The Applicant submits that such preprocessing of tasks into blocks of code prior to execution does not teach a person of skill in the art the recited job buffer that dynamically bundles jobs at runtime into tasks based on an equivalence of a job status of the jobs, where the job status includes the then-current value of a program counter and list of loop-counters. The teaching of Lorie is in the context of a system that operates differently than the apparatus of Claim 1 and, therefore, a person of

skill in the art would have neither motivation nor expectation of success in applying the teaching of Lorie to the system of Wilkinson, as proposed by the Office Action.

Fifth, the Office Action acknowledges Wilkinson as modified by Lorie does not disclose a job buffer dynamically bundling jobs into a task based on an equivalence of a job status of the jobs, but asserts that Kalman teaches such dynamic bundling, citing column 10, lines 9-26:

> It is common in a conventional RTOS to have, at any random time, groups of tasks in the same state, for example in the delayed state. Arrays and/or linked lists may be used to group the tasks for the purpose of accessing the highest-priority task of a given state as quickly as possible. Arrays consume relatively large amounts of memory but provide for random access to a given element, and hence are considered to be fast. Linked lists consume relatively little memory, but provide for linear access to a given element, and hence are considered to be slow. Elements of linked lists are linked together via pointers contained in the elements. Each element contains one or more pointers per list. Singly-linked lists use one pointer per element, doubly-linked lists use two, etc. The list itself may also have a handle, which often points to the list's first element (in singly-linked lists) or to an otherwise important element (in doubly-linked lists). Lists are often sorted based on one or more properties of the list elements, e.g. task priority.

That is, arrays or linked lists are used to order tasks having a common state based upon the tasks' priority. The purpose of such an array or list is to access the single highest-priority task of a given state as quickly as possible. As such, the purpose of the arrays/lists of Kalman are to separate tasks by priority for individual execution, rather than to group them for allocation together to an SIMD, as recited in Claim 1.

As such, Wilkinson, Lorie and Kalman fail to describe all the elements of Claim 1. Further, a person of skill in the art would have neither motivation nor reasonable expectation of success in combining the references as proposed by the Office Action. For at least these reasons, independent Claim 1 is patentable over the cited references. Independent Claim 15 recites elements analogous to

the novel and non-obvious elements of Claim 1 and, therefore, also is patentable over the cited references.

The Applicant submits that neither Pechanek nor Kim does anything to overcome the shortcomings of Wilkinson, Lorie and Kalman. Claims 2-6 and 9-14 depend from Claim 1, Claims 16-21 and 23-28 depend from Claim 15, and each includes all the elements of its respective base claim. Therefore, Claims 2-6, 9-14, 16-21 and 23-28 are patentable over the cited references.

Accordingly, the Applicant respectfully requests that the Examiner withdraw the § 103 rejection with respect to Claims 1-7, 9-21 and 23-28.

CONCLUSION

As a result of the foregoing, the Applicant asserts that the remaining claims in the Application are in condition for allowance, and respectfully requests an early allowance of such claims.

If any issues arise, or if the Examiner has any suggestions for expediting allowance of this Application, the Applicant respectfully invites the Examiner to contact the undersigned at the telephone number indicated below or at ***wmunck@munckcarter.com***.

The Commissioner is hereby authorized to charge any additional fees connected with this communication or credit any overpayment to Deposit Account No. 50-0208.

Respectfully submitted,

MUNCK CARTER, LLP

Date: Nov. 11, 2009

William A. Munck
Registration No. 39,308

P.O. Box 802432
Dallas, Texas 75380
(972) 628-3600 (main number)
(972) 628-3616 (fax)
E-mail: *wmunck@munckcarter.com*